JOURNAL OF VISUAL COMMUNICATION AND IMAGE REPRESENTATION
Vol. 8, No. 3, September, pp. 250–262, 1997
ARTICLE NO. VC970356

Motion-Adaptive Modelling of Scene Content for Very Low Bit Rate Model-Assisted Coding of Video

Wendi Rabiner*

Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

and

Arnaud Jacquin†

Multimedia Communications Research Laboratory, Lucent Technologies, Bell Laboratories, Murray Hill, New Jersey 07974-0636

Received January 5, 1996; accepted May 8, 1997

We describe a system which detects and tracks types of objects of interest specified *a priori*, such as human faces and bodies, in video sequences. Face location tracking algorithms described in previous documents are extended to enable robust and accurate tracking of faces and bodies in video scenes with *complex spatio-temporal backgrounds*, i.e. cluttered static backgrounds and moving backgrounds (for example, due to camera motion or zoom). The new tracking algorithm includes *background removal* obtained from global motion estimation (GME), as well as the use of *combined motion and edge data* and *knowledge-based temporal adaptation*, which jointly add significant robustness to the tracking. For typical "head-and-shoulders" video material with up to two persons in the scene and a still background, an additional 24% of successful tracking is achieved by the proposed algorithm, bringing the average success rate to about 96%. For more complex material with moving backgrounds, successful face and body tracking is achieved at an average rate of about 86%, whereas an algorithm which does not perform background removal could only achieve less than 10% of successful tracking. Initial coding experiments using the information obtained from face tracking for model-assisted coding of video in QCIF format at 16 kbps demonstrate the benefits of face tracking for very low bit rate coding of complex video material such as scenes with moving backgrounds. © 1997 Academic Press

1. INTRODUCTION

For very low bit rate coding applications, it is generally assumed that a video encoder should always benefit from knowledge of the content of the source video material, as opposed to performing "blind encoding" of pixel values or data blocks. This assumption was at the core of the concept of model-assisted coding of video teleconferencing sequences described in [1, 2], in which finer quantization (requiring the allocation of a higher coding rate) is performed in previously identified areas of interest, such as human faces.

The validity of this assumption is further demonstrated in the work described in this paper. Here, additional knowledge about sequence content in the form of global background motion estimation is obtained and used to perform background removal for more robust tracking of human faces and bodies. The added robustness is especially significant in cases of complex spatio-temporal scene backgrounds, which would typically occur with video data acquired from a hand-held video camera (e.g., in a mobile situation). Many foreground/background segmentation techniques have been proposed in the literature, initially in the relatively simple case of stationary (still) backgrounds [3, 4], then in the more general case of moving backgrounds [5–9]. In the former case, the techniques are typically based on frame differencing, followed by thresholding, and fail when the background is no longer stationary. In the latter case, where moving backgrounds occur due to, for example, camera motion or zoom, the techniques are more sophisticated and typically require a parametric modelling of the global background motion. We used the technique proposed by Tse *et al.* [5], which provides a good trade-off between computational complexity and performance.

In addition, our tracking algorithm follows a number of semantic rules imposed by scene content which make the tracking itself temporally adaptive and knowledge-based. These rules result in much higher tracking consistency than can be obtained with the purely spatial algorithm of [2].

* This work was carried-out while the author was a University Relations Intern supported by AT&T Consumer Products, Summer 1995.

† E-mail: arnaud@bell-labs.com.

1047-3203/97 $25.00

1/20

Therefore, knowledge about scene content is successfully used, not only by the video encoder, but also by the tracking module itself.

The organization of this paper is the following. Section 2 addresses global motion estimation and the subsequent foreground/background separation, as well as the generation of foreground motion-and-edge data to be used as input to the face and body tracking algorithm. Section 3 describes the tracking algorithm itself in detail. Section 4 shows tracking results in sequences with multiple persons in the scene and complex spatio–temporal backgrounds, as well as very low bit rate (16 kbps) coding results generated by a video coding platform which uses the face/body masks generated by the tracking algorithm. Section 5 concludes this paper by listing a number of applications for which robust object tracking in difficult environments can potentially be extremely useful.

2. SCENE BACKGROUND REMOVAL

2.1. Global Motion Estimation and Foreground/Background Separation

Global motion modelling and estimation is performed according to the technique proposed by Tse *et al.* described in [5]. This technique determines the *dominant global motion* in a scene, modelled as a combination of camera zoom—caused by a change of the focal length of the camera—and pan—caused by camera rotation about an axis parallel to the image plane. It is assumed that the motion of the scene background *is* the dominant motion—a very reasonable assumption for a wide range of video material—with the independent motion of foreground objects treated as uncorrelated noise. Objects with (local) motion vectors which are incompatible with the model of this dominant motion are therefore classified as "foreground objects."

Estimates of pan and zoom parameters are the result of an iterative algorithm which uses block-based motion vectors (d_i) obtained by "traditional" block-based motion estimation (ME) techniques. We used the full-search motion estimation algorithm with half-pixel accuracy described in ITU-T Recommendation H.263 [10], which provides 16 × 16 and 8 × 8 motion vectors in "advanced prediction" mode. Block-based ME is used to determine, for each block of the current frame, the best match in the previous frame. This *local* motion estimation can be modelled by the translational coordinate transformation for the pixels in block i:

$$U_i = U_i' + d_i, \tag{1}$$

where U_i' and U_i respectively denote the coordinates of a pixel in the current and previous frames. It is shown in [5] that global background motion from zoom and pan can be modelled by the coordinate transformation

$$\hat{U}_i = f_z U_i' + p, \tag{2}$$

where f_z is the camera zoom factor ($f_z > 1$ indicates that the camera is "zooming out"), p is a two-dimensional pan vector, and $\hat{U}_i$ is an estimate of the coordinates of a pixel in the globally motion-compensated block in the previous frame. This transformation is affine and can be written in matrix form as

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} f_z & 0 & p_x \\ 0 & f_z & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}. \tag{3}$$

Equivalently, Eq. (2) applied to the center pixel of block i provides an estimate $\hat{d}_i$ of the *global* motion vector for this block:

$$\hat{d}_i = (f_z - 1)U_i' + p. \tag{4}$$

The global zoom and pan parameters, unknown *a priori*, are approximated by iteratively minimizing an error metric E—the total l_2-norm of the difference between estimated (global) and "known" (local) motion vectors,

$$E = \sum_i \|e_i\|^2, \tag{5}$$

where

$$e_i = \hat{d}_i - d_i. \tag{6}$$

The minimization yields the estimates of zoom and pan at the kth iteration,

$$\hat{f}_{z_k} = \frac{\Sigma_i \langle U_i, U_i' \rangle - \frac{1}{N_k} \langle \Sigma_i U_i, \Sigma_i U_i' \rangle}{\Sigma_i \langle U_i', U_i' \rangle - \frac{1}{N_k} \langle \Sigma_i U_i', \Sigma_i U_i' \rangle}, \tag{7}$$

$$\hat{p}_k = \frac{1}{N_k} \left(\sum_i U_i - \hat{f}_{z_k} \sum_i U_i' \right), \tag{8}$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product. The initial estimates f_{z_0} and p_0 are computed with the summations taken over all N_0 local motion vectors.[1] For subsequent iterations ($k > 0$), only N_k blocks for which the magnitude of the error vector e_i is smaller than a threshold T_k are

[1] For input images in QCIF format (of size 176 × 144), N_0, the total number of 8 × 8 blocks is equal to 396 minus the number of blocks coded in INTRA mode; 16 × 16 blocks are treated as four 8 × 8 blocks with identical motion vectors.

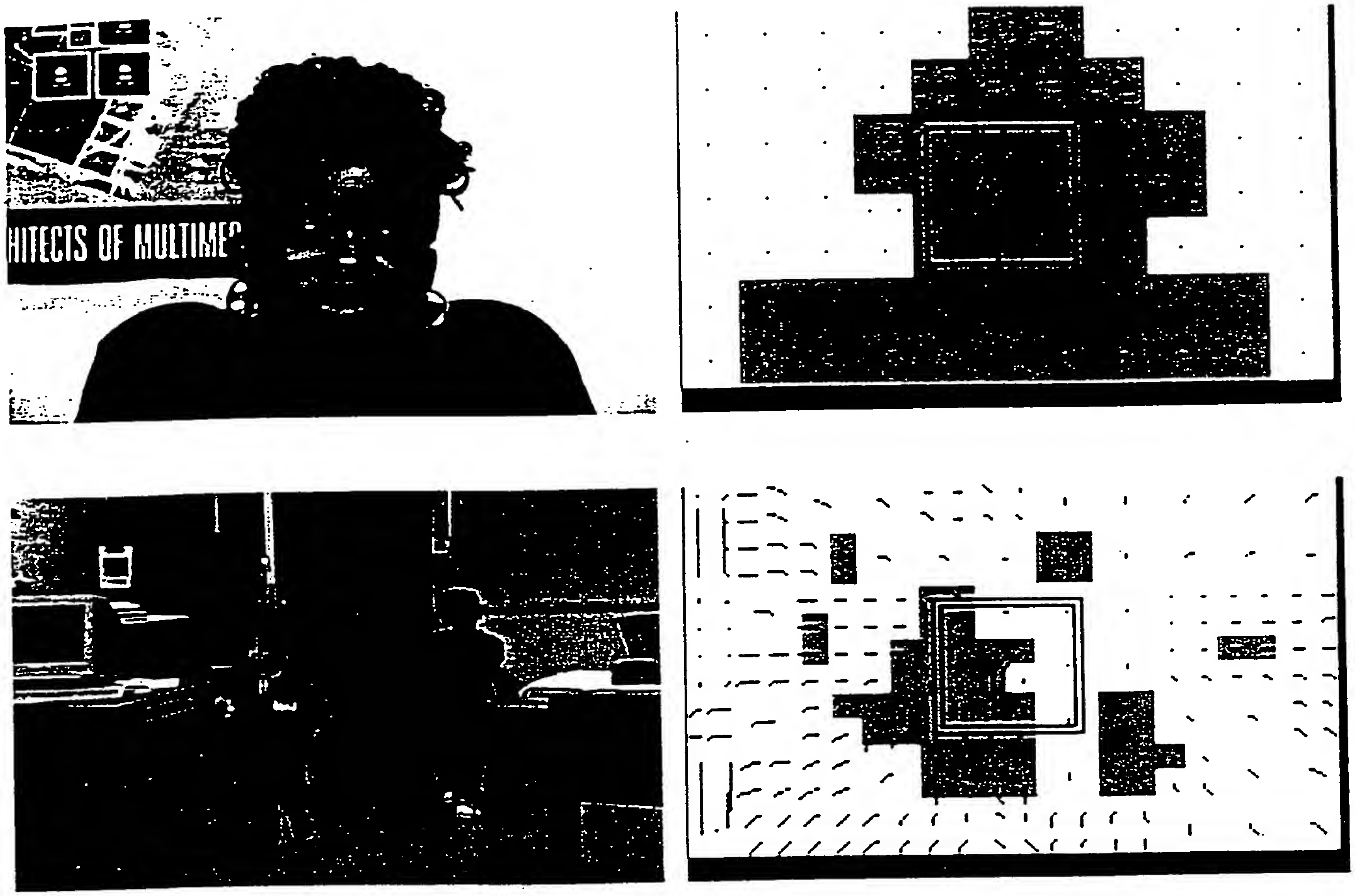

FIG. 1. Global motion estimation in sequences "Manya" (upper image pair) and "Sam." Left: Original CIF images. Right: Block-based motion vectors and foreground/background block segmentation (foreground blocks in grey).

used. This should indeed be done in order to prevent the biasing of the estimates by the independent motion of foreground objects and to progressively refine the global motion parameter estimates. In [5], Tse *et al.* do not specify how to select the parameters N_k and T_k. We chose to use a fixed number of three iterations in our experiments. The threshold T_k was chosen to decrease with iteration number according to the formula

$$T_k = T_{\max}/k \quad \text{for } k \in \{1, 2, 3\}, \tag{9}$$

where the upper-bound $T_{\max}$ of this threshold depends on scene complexity.[2] The final estimates f_z and p are obtained after three iterations, which we found to be enough to allow convergence of the estimates to an accurate description of the global motion while limiting the computational complexity.

Once the global motion of the scene has been estimated, a separation between scene background and foreground objects is obtained by comparing the final error between the local and global motion predictions. Blocks for which this error is smaller than a "tight" threshold T_{final} are classified as belonging to the scene background. T_{final} is based on the number of blocks that matched the background in the last iteration used to compute the global motion parameters (i.e., N_3), according to

$$T_{\text{final}} = \begin{cases} T_1, & \text{if } N_3 < \dfrac{N_0}{8}, \\ T_2, & \text{if } \dfrac{N_0}{8} \le N_3 < \dfrac{3N_0}{8}, \\ T_3, & \text{if } \dfrac{3N_0}{8} \le N_3 < \dfrac{5N_0}{8}, \\ T_4, & \text{if } N_3 \ge \dfrac{5N_0}{8}. \end{cases} \tag{10}$$

[2] For typical teleconferencing scenes with one or more people in the image frame, such as "Manya," "Mother–Child," and "Sam–Dad," $T_{\max}$ was chosen equal to 1 pixel. For more general scenes with complex moving backgrounds, such as the sequences "Foreman" and "Sam," $T_{\max}$ was chosen equal to 3 pixels.

9/20

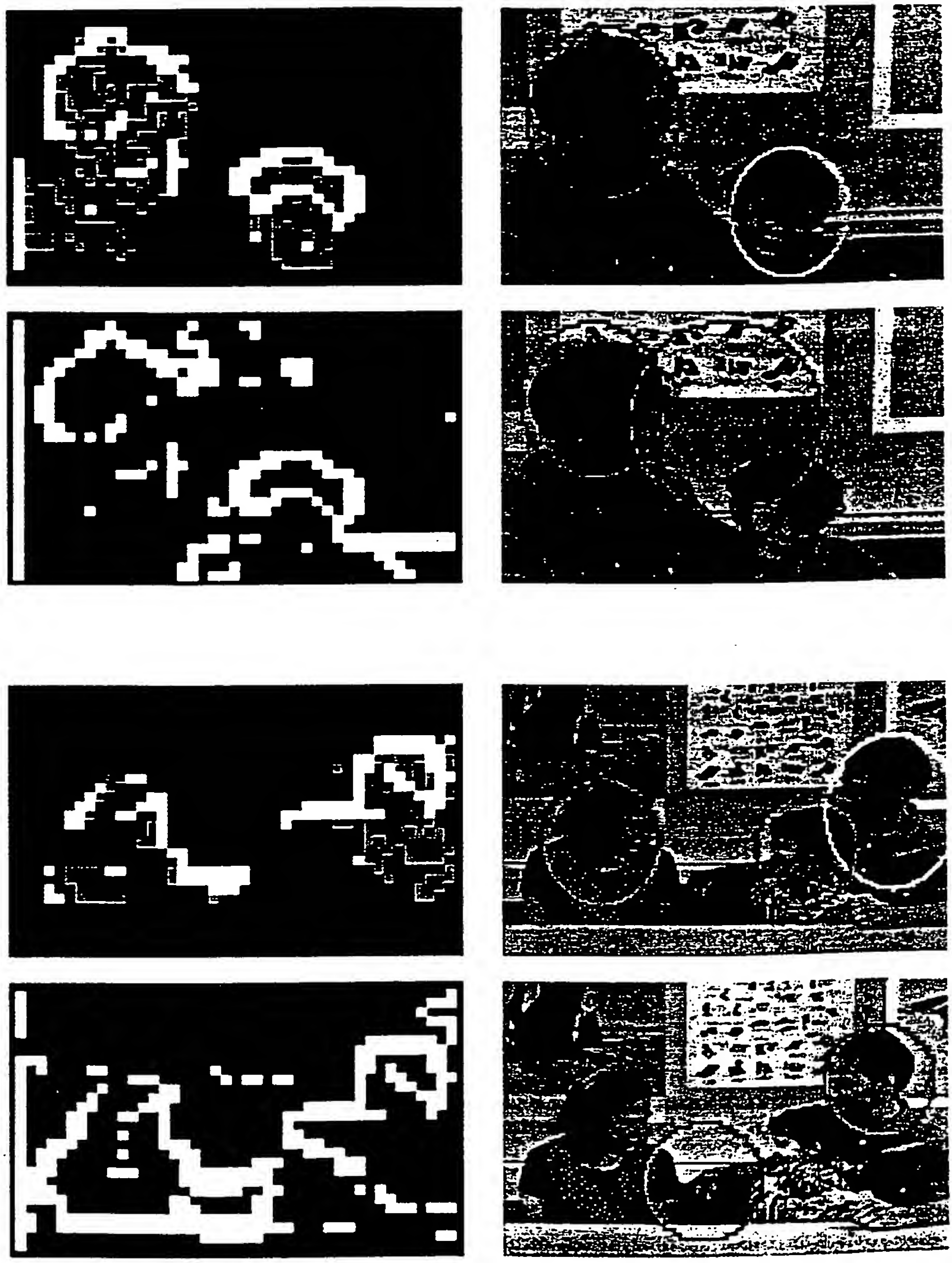

FIG. 3. Face tracking in sequences "Sam–Dad" (3a) and "Wendi–Sam–Dad" (3b). In each group of four images, the upper image pair corresponds to the new tracking algorithm with ternary downsampled motion-and-edge data (on left), and the lower image pair corresponds to the algorithm from [4] with binary downsampled edge data (on left). Face locations are overlayed on original QCIF images on the right.

This provides a method of constraining the number of background blocks not to be either excessively small or large.

The consistency of the foreground/background separation is further enhanced by nonlinear filtering. In particular, the label of foreground blocks is switched to "background" if these blocks are almost completely surrounded by background blocks[3] and vice versa. Moreover, blocks corresponding to areas uncovered by a pan or zoom-out are forced to be classified as background, regardless of their initial labelling.

Examples of foreground separation using this technique is shown in Fig. 1. The images on the right show local motion vectors (in grey, for 16 × 16 or 8 × 8 blocks), as well as the estimated global zoom and pan parameters.[4] Blocks classified as belonging to foreground objects are shown in grey; those belonging to the background are shown in white.

2.2. *Motion-Based Image Segmentation*

In [2], the data input to the head location detection algorithm was purely *spatial*, corresponding to edges and high-frequency textures of all objects, static or moving. Even though the algorithm is reliable in most head-and-shoulders video material, it can fail when the background is cluttered and/or moving (comparisons of algorithm performance are given in Section 4.1). In this section, we describe a motion-based image segmentation procedure which provides much more reliable input data where: (i) background edges and textures are erased; (ii) foreground objects appear in terms of both (spatial) edge data and (temporal) motion data. The contribution of foreground objects to the input data is therefore *enhanced*, which will make the tracking of these objects both easier and more robust.

The motion-based segmentation procedure is performed according to the block diagram of Fig. 2. A combined *motion-and-edge image* is created by overlaying a decimated edge image onto a globally motion-compensated decimated difference image (which we refer to as motion data). The edge image is the result of the decimation by 4 of an original QCIF luminance image, followed by the application of a classical Sobel edge detector. The motion data is created by globally motion-compensating an original QCIF luminance image using the estimated global motion parameters obtained in Section 2.1, followed by thresholding the error image.[5] Pixels with an absolute value larger than the threshold are considered foreground motion. The resulting binary globally motion-compensated difference image is then decimated by 4. A pixel in the decimated motion data image is classified as "foreground" if the number of foreground pixels in the corresponding 4 × 4 block before decimation is greater than a population threshold M.[6]

The decimated motion image and edge image are finally combined to create a *ternary* motion-and-edge image, where pixels can have one of the values:[7]

$$b_0 < b_1 < b_2. \tag{11}$$

Edge data pixels are first set to b_2, motion data pixels are then set to b_1 (unless they also belong to the edge data in which case they are left unchanged), and the remaining pixels are set to b_0 (the default value). Finally, data in areas classified as background as a result of the background classification stage described in Section 2.1 is "erased" (i.e., reset to b_0) in order to create a *foreground motion-and-edge image*.

Examples of such images are shown in the two upper-left quadrants of Figs. 3 and 4, where the three values b_0, b_1, b_2 correspond respectively to black, mid-grey, and white. It is interesting to note that in these images, the background has all but disappeared, even in the case of video sequences with very complex spatio-temporal backgrounds resulting from the combined effects of significant pan and background clutter (such as in "Foreman" in Fig. 4), or pan, zoom, and background clutter (such as in "Sam" also in Fig. 4). On the other hand, the purely spatial binary edge data that was used for head tracking in [2], shown in the lower left quadrants of Figs. 3 and 4, is visibly much less "clean."

3. FACE AND BODY TRACKING USING SEGMENTED IMAGES

3.1. *Fitness Metric*

The tracking algorithm uses foreground motion-and-edge images as input. The maximum number of objects (faces or human bodies) that the algorithm will find is specified *a priori*. The algorithm looks for "best elliptical fits" to the clumps of motion-and-edge data present in the input images. As was previously observed, ellipses provide a simple yet efficient way to capture locations of human heads. As a rough "first approximation," vertically ellon-

[3] This was defined specifically as either all nine neighboring blocks, or all nine but one of the four corner blocks.

[4] The black vector at the center of the image represents the global pan vector. A single black square (of width $w = 40$ pixels) represents a unit zoom factor. The second box, when present as in the "Sam" image, has width $w[1 - 2.5(f_z - 1)]$, i.e., proportional to the zoom factor. The image extracted from the "Sam" sequence is part of a zoom-out ($f_z > 1$).

[5] The threshold is fixed and was empirically chosen equal to 10.

[6] For typical teleconferencing scenes M was chosen equal to 4. For more general scenes, it was chosen equal to 8.

[7] We used the numerical values: $b_0 = 0$, $b_1 = 128$, and $b_2 = 255$.

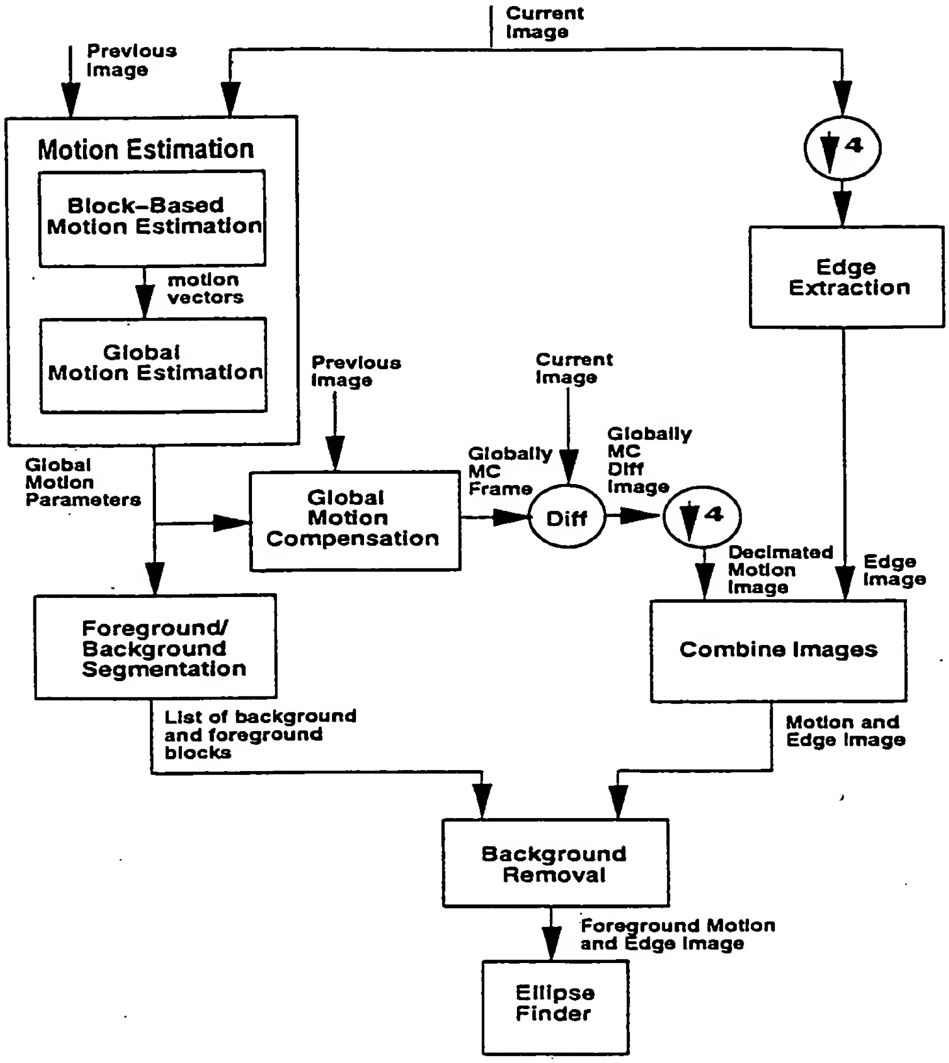

FIG. 2. Generation of foreground motion-and-edge images using block-based and global motion estimation.

gated ellipses can also be used to capture the outline of a standing human body.

We define a "fitness metric" F as the sum of three quantities:

$$F = d_{\text{border}} + d_{\text{motion}} + P_{\text{motion}}, \qquad (12)$$

where d_{border} indicates the density of edge data on a candidate ellipse border, d_{motion} indicates the density of motion data inside the ellipse, and P_{motion} indicates the percentage of motion data inside the ellipse (relative to a window of width one-half of the input image width, centered on the ellipse). Each of the terms that constitute F are nonnegative and upper-bounded by one, so that its maximum value $F_{\max}$ is equal to 3; d_{border} recalls the fitness ratio of [2]—its purpose is to measure the contribution of edge data shaped as elliptical arcs.[8] The sum $d_{\text{motion}} + P_{\text{motion}}$ measures the contribution, in both an absolute and a relative sense, of motion data organized in elliptical clumps.

3.2. *Knowledge-Based Temporal Adaptation and Learning of Scene Content*

In addition to computing a fitness measure for the detection of objects of interest, the algorithm follows a number of "semantic rules" naturally imposed by scene content.

The algorithm uses the previous best-fitting ellipses and transforms them under the affine zoom and pan transformation of Eq. (3) to obtain a prediction of where these ellipses should be in the current frame if the motion-and-

[8] Candidate ellipses are required to have a d_{border} of at least 0.2.

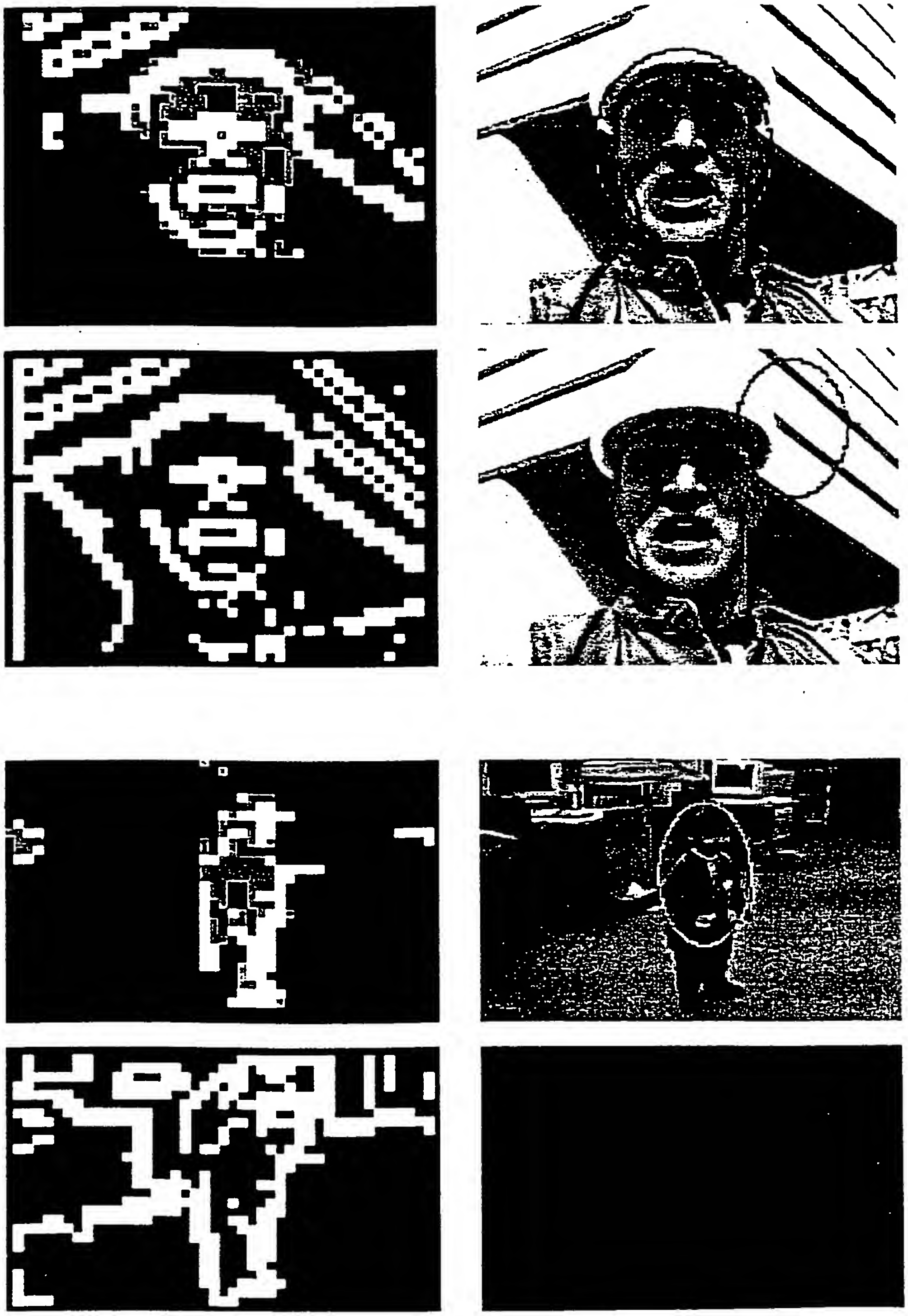

FIG. 4. Face tracking in sequences "Foreman" (4a) and "Sam" (4b). In each group of four images, the upper image pair corresponds to the new tracking algorithm with ternary downsampled motion-and-edge data (on left), and the lower image pair corresponds to the algorithm from [4] with binary downsampled edge data (on left). Face locations are overlayed on original QCIF images on the right.

13/20

6

edge data disappears. This corresponds to a situation where the face/person has stopped moving (in case of a still background), or, more generally, is moving "in sync" with the moving background. For an ellipse $\mathscr{E}$ in the previous frame, with cartesian equation,

$$ax^2 + 2bxy + cy^2 + 2\,dx + 2ey + f = 0, \tag{13}$$

which can equivalently be written in matrix form,

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}^{\mathrm{T}} \begin{bmatrix} a & b & d \\ b & c & e \\ d & e & f \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0; \tag{14}$$

the equation of the transformed ellipse $\mathscr{E}'$ is simply obtained by

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}^{\mathrm{T}} \begin{bmatrix} f_z & 0 & 0 \\ 0 & f_z & 0 \\ p_x & p_y & 1 \end{bmatrix} \begin{bmatrix} a & b & d \\ b & c & e \\ d & e & f \end{bmatrix} \begin{bmatrix} f_z & 0 & p_x \\ 0 & f_z & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = 0. \tag{15}$$

Besides, the algorithm has *memory* and keeps track of the following information from the previous frame: (i) the location of objects of interest; (ii) the size and shape of these objects (i.e., major axis and aspect ratio for elliptical object models); and (iii) the number of objects of interest, including possible occlusions of one object by another and persons entering and leaving the scene. Based on this information, the algorithm adapts the search range of the ellipse parameters in the current frame.

If the maximum number of fits was not found for the previous frame, the entire range of ellipse parameters is searched to determine the presence of the other objects of interest. In order for the algorithm to determine that another object of interest has entered the scene, the object must be far enough away from all the other objects that the algorithm is certain a new, distinct object has been found. To determine occlusions of one object by another, the algorithm checks the separation between the centers of the ellipses. If this separation is small, indicating occlusion, one of the ellipses (the smaller of the two) is removed, as only one ellipse is necessary to track the two objects. Once the two objects separate, the algorithm again tracks the previously occluded object. Similarly, the algorithm determines that an object is leaving the scene when the center of the predicted object localization is close to any edge of the scene. If this occurs, the ellipse is removed and the algorithm determines that there is one fewer object of interest present. If the object returns to the scene, the algorithm will start again tracking its motion. An algorithmic description of these concepts is provided in the next section.

The use of the combination of the above-mentioned semantic rules makes the tracking algorithm both *temporally adaptive* and *knowledge-based*, in that it continuously updates its knowledge of scene content and continuously makes decisions based upon this knowledge.

3.3. *Final Ellipse Selection*

The final ellipse selection algorithm relies jointly on measures of fitness and on the semantic rules described in Section 3.2. The algorithm is shown in Fig. 5. It uses two *fitness thresholds*, C_1 and C_2, which satisfy:

$$0 < C_2 < C_1 < F_{\max} \tag{16}$$

and differenciate "very high" and merely "high" fitness measures.[9] It also uses two *separation thresholds*, D_1 and D_2, which satisfy

$$D_2 < D_1. \tag{17}$$

These separation thresholds limit the magnitude of the motion between ellipse centers in two consecutive frames.[10] This is done in order to make tracking consistent from frame to frame.

For each object of interest, the selection algorithm goes through an ordered list of candidate ellipses as shown in Fig. 5. The first choice, when available, corresponds to a "high" fitness value ($C > C_2$), and small separation between the centers of the previous and current face/body

[9] We chose $C_1 = 1.6$ and $C_2 = 1.4$ in our tracking experiments.

[10] They should therefore be chosen to depend on the temporal sampling rate of the video input—the higher the frame rate, the smaller the thresholds. We used the empirically derived formulas for these parameters,

$$D_1 = (f_s/3) + 8, \quad D_2 = (f_s/3) + 2, \tag{18}$$

where f_s denotes the number of frames skipped in a video sequence initially sampled at 30 fps to obtain a video input downsampled to a constant F fps (typically 5 fps).

FIG. 6. Stills from QCIF sequence "Manya" coded at 16 kbps, constant 5 fps frame rate, without (left), and with (right), model-assisted rate control.

FIG. 7. Stills from QCIF sequence "Sam–Dad" coded at 16 kbps, constant 5 fps frame rate, without (left) and with (right) model-assisted rate control.

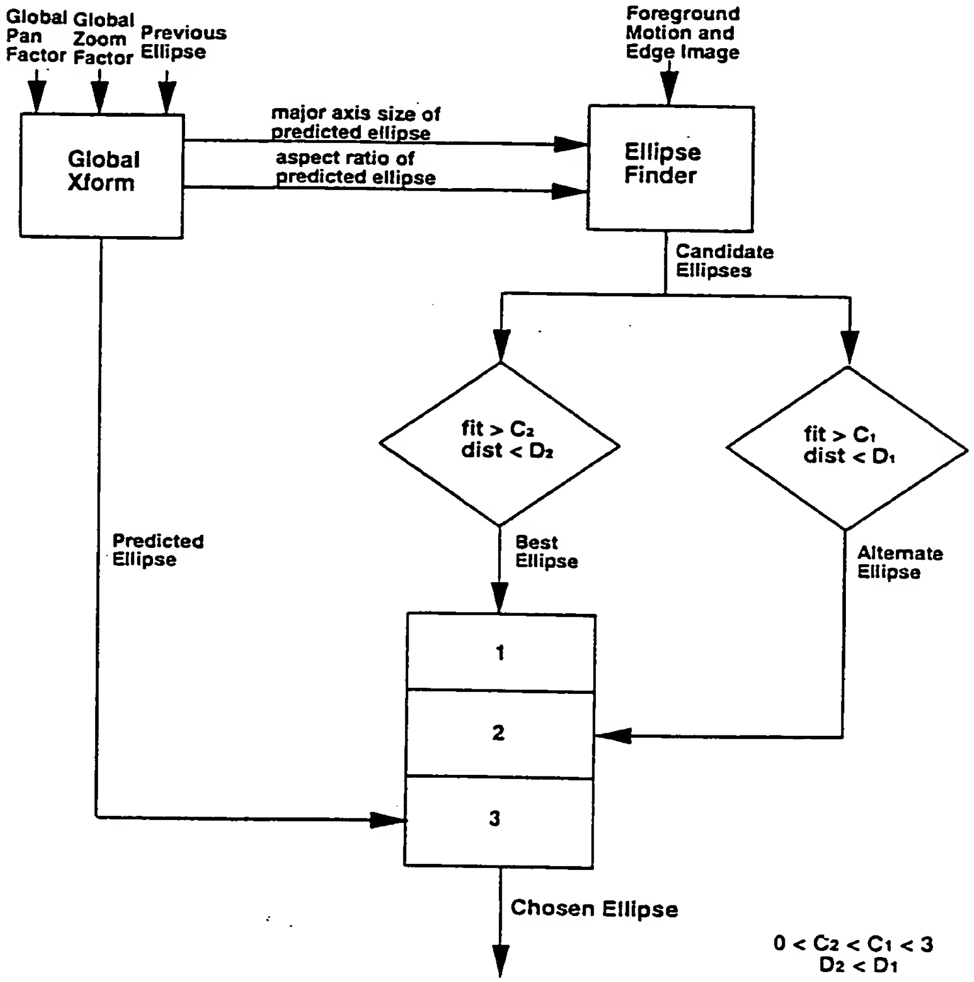

FIG. 5. Ellipse selection algorithm.

locations ($D < D_2$)—the combination which the algorithm chooses with highest confidence. The second choice, when available, corresponds to "very high" fitness value ($C > C_1$) and "not-too-large" separation between locations ($D < D_1$). The third choice is the predicted ellipse, which is available only if an object of interest was present in the previous image.

Elliptical object locations selected in this fashion are shown in the upper-right quadrants of Figs. 3 and 4, overlayed onto original images in QCIF format.

4. RESULTS

4.1. *Tracking Results*

Tables 1 and 2 show the robustness of both the "old tracking" of [2], and the "new tracking" algorithm of Section 3. In Table 1, percentages of successful tracking are given for a few typical head-and-shoulders test sequences: "Manya" (15 s), "Mother–Child" (6.5 s), and "Sam–Dad" (25 s). In Table 2, percentages of success are given for the significantly more complex sequences "Foreman" (13 s) and "Sam" (10 s).

False alarms were defined to be cases where the algorithm determined that a face/body was present when it was not, and *missed objects* to be cases where either: (i) a face/body was present and not detected, (ii) the tracking missed part or all of the face,[11] or (iii) the algorithm picked up a "very large" region, which happens to encompass face regions, but is too large to be useful to the encoder. The second set of numbers, given in the lower parts of the tables, corresponds to not counting missed objects *before* the person began moving his/her head, which can be dismissed as a transient phase.

[11] Missed eyes or mouth resulted in a *missed objects* label.

TABLE 1
Robustness of New Face Tracking Algorithm for Head-and-Shoulders Scences

	Old tracking			New tracking		
Sequence name	False alarms	Missed faces	Correct tracking	False alarms	Missed faces	Correct tracking
Manya	0	4.5	95.5	0.0	1.5	98.5
MotherChild						
Mother	0.0	1.7	98.3	0.0	2.0	98.0
Child	0.0	0.0	100.0	0.0	2.0	98.0
SamDad						
Sam	0.0	29.9	70.1	0.0	0.5	99.5
Dad	0.0	40.8	59.2	0.0	20.2	79.8
MotherChild						
Mother	0.0	1.7	98.3	0.0	0.0	100.0
Child	0.0	0.0	100.0	0.0	0.0	100.0
SamDad						
Sam	0.0	29.9	70.1	0.0	0.5	99.5
Dad	0.0	40.8	59.2	0.0	10.0	90.0

For the sequences of Table 1, with up to two persons in the scene and a still background, an additional 24% of successful tracking is achieved with the new algorithm, bringing the average successful head tracking rate[12] to about 96%. For the more complex sequences of Table 2, which have complex moving backgrounds, successful head tracking is achieved at an average success rate of about 86%, whereas the old algorithm, which does not perform background removal, only achieves a success rate less than 10%. These examples clearly illustrate the dramatic improvements achieved by the new algorithm.

It was also observed that tighter localization of faces was obtained with the new face tracking algorithm. Moreover, this new algorithm runs approximately two orders of magnitude faster on an SGI Indigo workstation (a few seconds per frame as opposed to a few minutes with the old algorithm). This can be explained by a combination of reasons: (i) cleaner input data which limits the search for ellipses, (ii) a reduction of the search space of ellipse parameters due to temporal adaptation, and (iii) a simpler fitness metric.

4.2. *Model-Assisted Coding*

Coding simulations were performed using the software platform described in [11]. The core coding platform is based on ITU-T Recommendation H.263 [10], with a number of additional features such as prefiltering, face tracking, adaptive VQ, model-assisted rate control, and adaptive

TABLE 2
Robustness of New Face Tracking Algorithm for Scenes with Complex Moving Backgrounds

	Old tracking			New tracking		
Sequence name	False alarms	Missed faces	Correct tracking	False alarms	Missed faces	Correct tracking
Foreman	22.5	68.5	9.0	6.0	10.5	83.5
SamPrash				1.3	18.0	80.7
Foreman	22.5	68.5	9.0	6.0	3.5	90.5

[12] Average success rates are computed with the numbers in the lower parts of the tables weighted by sequence length.

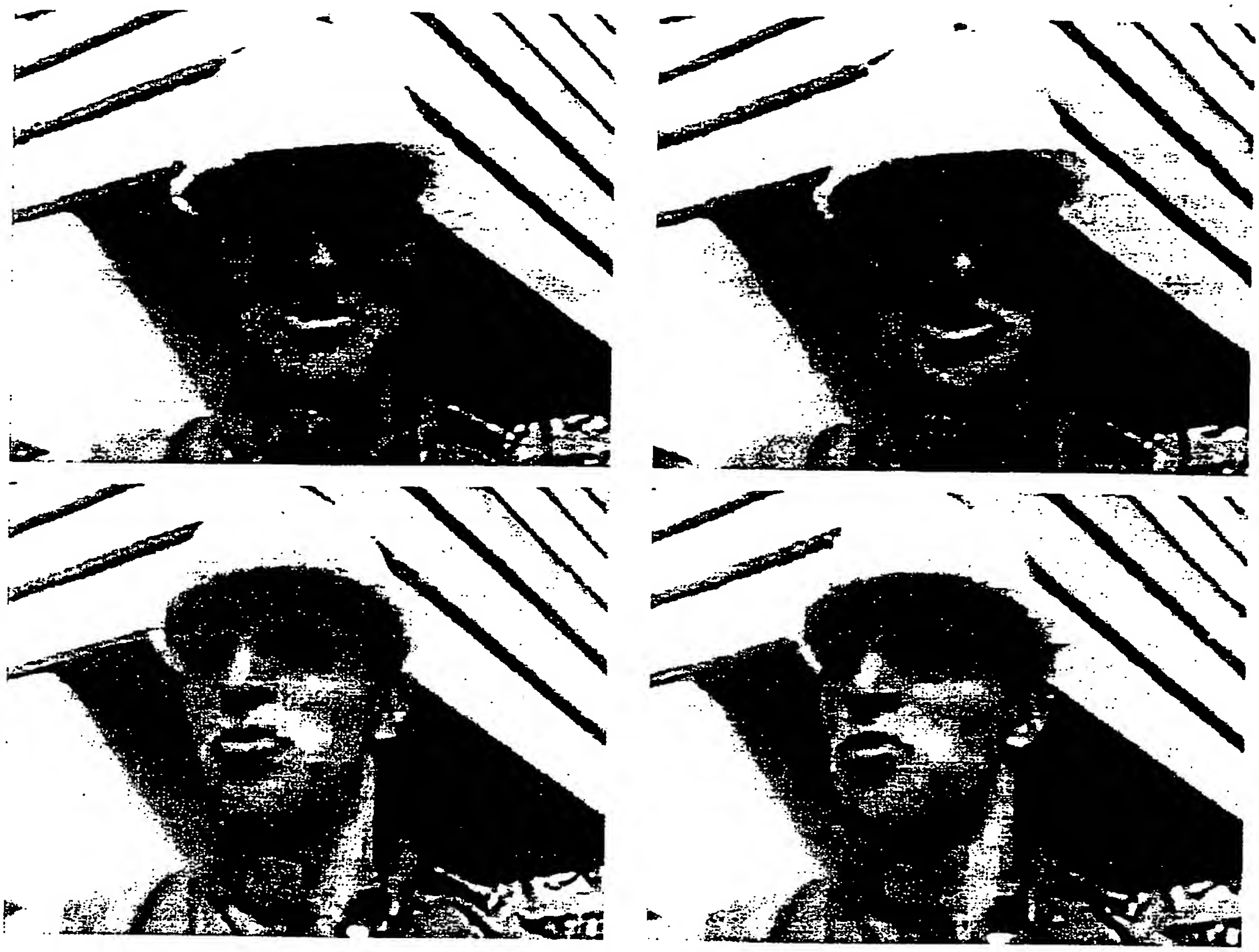

FIG. 8. Stills from QCIF ITU-T sequence "Foreman" coded at 16 kbps, variable frame rate (average: 2.5 fps), without (left) and with (right) model-assisted rate control.

postfiltering. Of particular relevance here is the rate control module which uses face tracking results to identify regions of interest in the image—faces for video teleconferencing scenes—and allocates a higher coding rate to these areas by diverting bits from the remaining areas (clothing, background). The increase in subjective quality obtained from using this type of model-assisted paradigm was studied in [12]. The platform can be run at a number of different coding rates and video input resolutions. It can also operate under different modes, in particular, constant frame rate output (CFR) and variable frame rate output (VFR) for challenging material such as scenes with complex moving backgrounds.

Coding results are shown in Figs. 6–8 for the sequences "Manya," "Sam–Dad," and "Foreman," respectively. All three sequences were encoded at a total coding rate of 16 kbps. "Manya" and "Sam–Dad" were coded at a fixed frame rate of 5 fps, whereas "Foreman" was coded at a variable frame rate (the average frame rate over the sequence was 2.5 fps). For the sequence "Sam–Dad," the additional localization of rectangular "eyes–noise–mouth" regions was performed as described in [2] and used, which results in improved coding of facial features only for people facing the camera (the child in this case). For the sequences "Manya" and "Foreman" the improvement of facial features when face tracking is performed and used by the model-assisted coder is particularly visible (see the images on the right in Figs. 6 and 8).

5. CONCLUSIONS AND APPLICATIONS

We described a modular system which uses scene background removal to robustly detect and track locations of faces and bodies of people in video sequences with complex

18/20

spatio-temporal backgrounds. The algorithms can be advantageously used in a large number of applications such as: (i) model-assisted low-bit-rate coding of video with complex backgrounds and multiple people in the scene, (ii) surveillance systems to track people in motion and monitor intrusions, and (iii) content-based indexing of video databases, as a preprocessing stage for face recognition.

For the first application areas, we showed video coding results which illustrate the advantage of using *knowledge* about scene content for very low bit rate applications. In the case of video sequences where the background is moving, which would typically be the case in *wireless video telephony*, face tracking remains quite robust, with average successful head tracking rates of about 86%. The usefulness of the information obtained from face tracking in this case was illustrated by coding results at the very low bit rate of 16 kbps, where a model-assisted coding system produces images with significantly better-rendered facial features.

ACKNOWLEDGMENTS

The authors thank James Pawlyk and Keith Bauer for their assistance in the acquisition of the video material used in this work.

REFERENCES

1. A. Eleftheriadis and A. Jacquin, Automatic face location detection and tracking for model-assisted coding of video teleconferencing sequences at low bit rates, *Signal Process. Image Commun.* 7, No. 3, 1995, 231–248.
2. A. Eleftheriadis and A. Jacquin, Automatic face location detection for model-assisted rate control in H.261-compatible coding of video, *Signal Process. Image Commun.* 7, Nos. 4–6, 1995, 435–455.
3. C. Lettera and L. Masera, Foreground/background segmentation in videotelephony, *Signal Process. Image Commun.* 1, No. 2, 1989, 181–189.
4. W. Guse, M. Gilge, and B. Hürtgen, Effective exploitation of background memory for coding of moving video using object mask generation, *Proc. SPIE VCIP '90* **1360**, 1990, 512–523.
5. Y. T. Tse and R. L. Baker, Global zoom/pan estimation and compensation for video compression, in *Proc. ICASSP '91*, pp. 2725–2728.
6. D. Adolph and R. Buschmann, 1.15 Mbit/s coding of video signals including global motion compensation, *Signal Process. Image Commun.* 3, Nos. 2-3, 1991, 259–274.
7. A. Amitay and D. Malah, Global-motion estimation in image sequences of 3-D scenes for coding applications, *Signal Process. Image Commun.* 6, No. 6, 1995, 507–520.
8. C. Swain and T. Chen, Defocus-based image segmentation, in *Proc. ICASSP '95* pp. 2403–2406.
9. F. Moscheni, F. Dufaux, and M. Kunt, A new two-stage global/local motion estimation based on a background/foreground segmentation," *Proc. ICASSP '95*, pp. 2261–2264.
10. Draft recommendation H.263: Video coding for narrow telecommunication channels, Boston, June 1995.
11. J. Hartung, A. Jacquin, H. Okada, and J. Rosenberg, Object-based H.263 compatible video coding platform for conferencing applications, to appear. [JSAC special issue on very low bit-rate video coding]
12. A. Jacquin, Perceptual quality evaluation of low bit rate model-assisted video, in *Proc. International Symp. on Multimedia Communications and Video Coding, New York, October 1995*, pp. 285–291.

ARNAUD JACQUIN (Member, IEEE) was born in Reims, France, in 1964. He received the Diplôme d'Ingénieur in Electrical Engineering from Ecole Supérieure d'Electricité, Gif-sur-Yvette, France in 1986. He received the M.S. degree in Electrical Engineering in 1987, and the Ph.D. degree in Mathematics in 1989, both from the Georgia Institute of Technology, in Atlanta, Georgia. Since 1990 he has been a Member of the Technical Staff at AT&T Bell Laboratories (now Lucent Technologies, Bell Laboratories), in Murray Hill, New Jersey, in the Multimedia Communications Research Laboratory, where he has been engaged in research on low bit rate digital video coding for teleconferencing/videotelephony. His research interests are in the areas of image and video compression, fractals, computer vision, and computer graphics. He is the recipient of the IEEE Signal Processing Society's 1993 Senior Award in the Multidimensional Signal processing Area for his paper on "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations," which appeared in the January 1992 issue of the *IEEE Transactions on Image Processing*.

WENDI RABINER received the B.S. degree in Electrical Engineering from Cornell University in 1995. She is currently a graduate student working on her M.S. degree at the Massachusetts Institute of Technology in the Research Laboratory of Electronics. Her research interests include image processing and coding, specifically algorithms designed for low power video systems. She is a member of Eta Kappa Nu, Tau Beta Pi, and the IEEE.